

Aplicación Chat sobre una red DTN

Javier Villar Almirón

Resumen— El uso de dispositivos móviles es cada día mayor, al igual que las aplicaciones destinadas a establecer comunicaciones entre los usuarios. Las nuevas tecnologías de red permiten desarrollar nuevas formas de comunicación sobre las mismas. Este proyecto se ha realizado con el objetivo de desarrollar una aplicación móvil para la plataforma Android con la finalidad de intercambiar mensajes entre los usuarios dentro de una red específica. Dicha aplicación trabajaría sobre una red tolerante a retardos (DTN) la cual se encontraría dentro del campus de la Universidad Autónoma de Barcelona. La finalidad de los mensajes de la aplicación es la de comunicar a otros usuarios, que dispongan de la aplicación, de alguna actividad, como puede ser: una asamblea, una conferencia, etc; y que se realice en alguna de las facultades, por ello la aplicación se divide en salas/grupos según la zona/facultad a la que se hace referencia. Los mensajes que circulan por la red son tratados y almacenados por una de las máquinas conectadas a la red la cual tiene una infraestructura LAMP (Linux, Apache, MySQL y PHP).

Palabras clave— aplicación móvil, Android, mensajes, red tolerante a retardos(DTN), infraestructura LAMP.

Abstract— The use of mobile devices is growing daily, as well as applications designed to establish communication between users. New network technologies ease the development of new ways of communication about these technologies. This project has aimed to develop a mobile application for Android with the objective of exchanging messages among users sharing a particular network. This application would work in a Delay/Disruption Tolerant Network (DTN), which could be found in the Universidad Autónoma de Barcelona's campus. The purpose of the aforesaid messages is that of communicating other users about some activity, such as: an assembly or a conference, among others. These events must take place in any of the university faculties; therefore, the application is divided in different groups depending on the area/faculty to which those messages make reference. The messages that circulate through the network are managed and stored by one of the machines connected to the network, which has a LAMP (Linux, Apache, MySQL y PHP) infrastructure.

Index Terms— mobile application, Android, messages, Delay/Disruption Tolerant Network (DTN), LAMP infrastructure.

1 INTRODUCCIÓN

La finalidad de este artículo es la de exponer el trabajo realizado a la hora de crear una aplicación móvil orientada a la comunicación y que se desarrolla sobre una red tolerante a retardos.

El contenido de este artículo se estructura de la siguiente forma: en primer lugar se definirán los objetivos que marcarán el desarrollo del trabajo.

En segundo lugar se expondrá el estado del arte en relación al ámbito del trabajo.

En tercer lugar se explicará la metodología de trabajo con las diferentes partes que engloban este proyecto.

En cuarto lugar se hará una valoración de los resultados obtenidos a lo largo del desarrollo del proyecto con una valoración de los mismos.

Finalmente se expondrán una serie de conclusiones extraídas del trabajo y unas líneas de continuación para un trabajo futuro.

- E-mail de contacto: Javier.villaralm@e-campus.uab.cat
- Mención realizada: Tecnologías de la Información.
- Trabajo tutorizado por: Guillermo Navarro Arribas (Departamento de Ingeniería de la Información y de las Comunicaciones)
- Curso 2014/15

2 OBJETIVOS

La realización de este trabajo contempla 2 objetivos principales que a su vez implican diversos subobjetivos a realizar.

El primer objetivo a realizar se trata del diseño de una aplicación móvil, para la plataforma *Android*, la cual deberá ser capaz de conectarse a una red que le ha sido indicada previamente, y también deberá ser capaz de enviar mensajes a un servidor que se encuentra dentro de la red y a su vez, poder recoger mensajes del mismo. Este objetivo se puede fraccionar en 2 subobjetivos, que son:

- Elaboración de la parte funcional del aplicativo, que recoge las funcionalidades básicas de la aplicación (conectarse a la red, envío/recepción de mensajes, etc),
- Diseño y creación de la interficie gráfica de la aplicación.

El segundo objetivo a completar, es el de la creación de una infraestructura de red sobre la cual trabajará el aplicativo que se haya creado anteriormente. Dicha infraestructura debe tener uno o más puntos de acceso a la red de manera que los terminales móviles puedan conectarse

a ella y así establecer las posibles comunicaciones. Dentro de la red, habrá nodos que se encargaran de mover los datos des del servidor hasta los puntos de acceso, y vice-versa. Estos nodos tienen la capacidad de retener la información hasta que les sea posible distribuirla a otro nodo, gracias a la implementación de un protocolo DTN (*Delay/Disruption Tolerant Network*). Este objetivo también se fragmenta en varios subobjetivos, que son:

- Creación del punto de acceso o *hotspot*.
- Creación del servidor web para la gestión y el almacenamiento de mensajes.
- Creación de varios dispositivos que implementen el protocolo DTN para la comunicación entre el servidor y los puntos de acceso.

Al comienzo del proyecto se determinó el escenario de red concreto que simula una red dentro del Campus de la UAB(Universidad Autónoma de Barcelona) y que a su vez hace de ejemplo del tipo de red sobre el que se moverá la información, como se puede observar en la Fig.1.

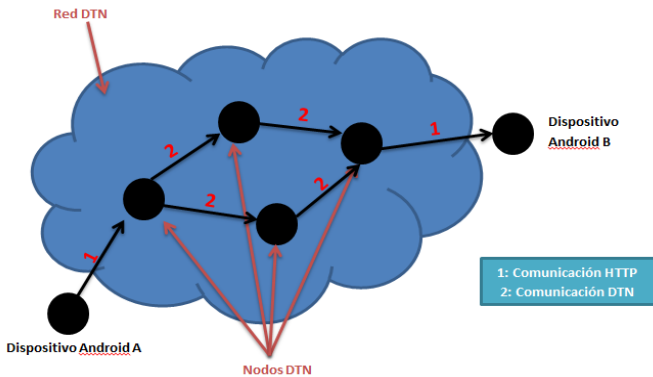


Fig.1: Escenario de red.

Por otro lado, teniendo en cuenta que se trata de una aplicación relacionada con el Campus de la universidad, se decidió dar un sentido al aplicativo más allá del simple hecho de comunicarse con ella. El objetivo que se incorporó fue el de que la finalidad de los mensajes que circularán por la red sea el de anunciar acontecimientos que se produzcan en el propio Campus. Por ese motivo se decidió también, dividir la aplicación en distintas salas/grupos que hacen referencia a la facultad sobre la cual tiene lugar el acontecimiento que se quiera comunicar (p.e. charlas divulgativas, conferencias, asambleas, etc). Por lo que respecta a los datos, cada mensaje que se difunda solo debe contener la facultad a la que se hacer referencia y el cuerpo del mensaje, también se crea información importante para la gestión de los mensajes pero ajena al usuario final, como pueden ser: fecha de creación del mensaje y un identificador del mismo.

3 ESTADO DEL ARTE

En el mundo actual cada vez son más los dispositivos tecnológicos que necesitan estar conectados a la red. En muchas ocasiones no es necesario que dichos dispositivos

tengan acceso a Internet pero sí que en numerosas ocasiones necesitan comunicarse con otros dispositivos y para ello necesitan una red común donde intercambiar la información. El número de dispositivos móviles en las sociedades actuales ha incrementado de forma drástica y además los dispositivos móviles de hoy en día, ya son capaces de hacer las mismas funcionalidades que los ordenadores de no mucho tiempo atrás. La combinación de ambos aspectos ha hecho que el mundo de las Tecnologías de la Información tenga que avanzar a pasos gigante para cubrir las demandas que se presentan en el mundo actual.

Las redes tolerantes al retardo tienen la capacidad de funcionar en terrenos hostiles donde no siempre hay un camino entre dos puntos de la red que se quieren comunicar. Para ello los nodos que forman la red DTN disponen de una unidad de almacenamiento persistente para guardar la información hasta el momento en que esta pueda ser enviada a otro nodo de la red, esto implica que los nodos pueden actuar como enrutadores y como hosts. El protocolo DTN trabaja sobre los distintos protocolos de red ya existentes gracias a la inserción de una capa llama "capa de fardos", situada entre la capa de transporte y la capa de aplicación, que permite homogeneizar redes con distinta naturaleza.

Lo que empezó siendo una solución para las comunicaciones interplanetarias, el protocolo DTN ofrece muchas posibilidades para las comunicaciones en zonas donde es difícil implementar una gran estructura de red por la composición del propio terreno o incluso para zonas donde el poder adquisitivo es bajo y no se puede montar una infraestructura de red idónea. También ofrece solución en aquellos casos donde la comunicación se produce en entornos hostiles, de manera que no siempre hay un camino disponible des del nodo emisor hasta el nodo receptor.

El desarrollo de aplicaciones destinadas a trabajar sobre una red tolerante al retardo está poco trabajado actualmente ya que el objetivo no es realizar nuevas aplicaciones para esta nueva tecnología de red, si no que la nueva tecnología de red sea lo suficientemente eficaz para las aplicaciones actuales que siguen el paradigma de red clásico. Por otra parte existen aplicaciones que habilitan la "capa de fardos" en la plataforma sobre la que han sido desarrolladas (p.e: Linux, Android, etc), con el objetivo de que los dispositivos con dicho software actúen como nodos en una red DTN y pasar a formar parte de la misma.

Por otra parte también cabe destacar que si se produce un desarrollo eficiente de esta tecnología, muchas aplicaciones que requieran de la comunicación con otros dispositivos, pero no de manera inmediata, podrán cumplir su función sin la necesidad de estar conectados a una infraestructura concreta, simplemente incorporando en los distintos dispositivos esta tecnología.

4 METODOLOGÍA

A continuación se detallarán los diversos pasos que se han seguido para la realización del proyecto, ordenados en apartado según su funcionalidad dentro del propio

proyecto.

4.1 Análisis de requisitos

Como ya se ha comentado en el apartado de “Objetivos”, inicialmente los requisitos que había para el desarrollo del proyecto eran pocos pero que implicaban una forma de desarrollo del proyecto específica. Los únicos requisitos iniciales que se tenían en cuenta eran: poder comunicarse entre los usuarios que disponen de la aplicación y están conectados a la red; y que la red en cuestión sea tolerante a retardos.

El primer requisito que afloró, una vez se empezó a analizar las necesidades que incorporaban los requisitos iniciales, fue el de la necesidad de crear un punto de acceso (*Hotspot*) para que los dispositivos *Android* puedan acceder a la red. Otro requisito necesario fue el de incorporar un servidor dentro de la red, para que este pueda gestionar y almacenar los mensajes que se difunden por la ella. Se introdujo la distribución de los mensajes en salas/grupos como requisito para el aplicativo, con la finalidad de dar un sentido concreto a dicho aplicativo ya que la definición de chat dejaba mucho margen de maniobra en este aspecto.

4.2 Implementación.

Una vez ya se tenía constancia de los requisitos mínimos para poder empezar con el trabajo se procedió a ello como se explicará a continuación.

4.2.1 Desarrollo del aplicativo móvil

Lo primero e indispensable para este proyecto era tener una aplicación móvil que cumpliera las necesidades que emanaban de los requisitos acordados.

Lo primero que se realizó fue crear una vista o *layout*, inicial, simple la cual contiene el logo de la aplicación y un botón que al ser presionado llamará a la función “*establecerConexion()*”, que tiene como objetivo conectarse a un punto de acceso con una *SSID*(*Service Set Identifier*) y una contraseña específicos; y en el caso de que se haya podido establecer la conexión abrir la 2ª vista/*layout* del aplicativo o, en el caso de que no se haya podido establecer la conexión, informar de no se ha podido conectar con el punto de acceso y mantener la vista inicial (ver las figuras de A2 en el apéndice). En la Fig.2 se puede ver la vista principal en el momento en el que no se ha podido establecer la conexión con la red.

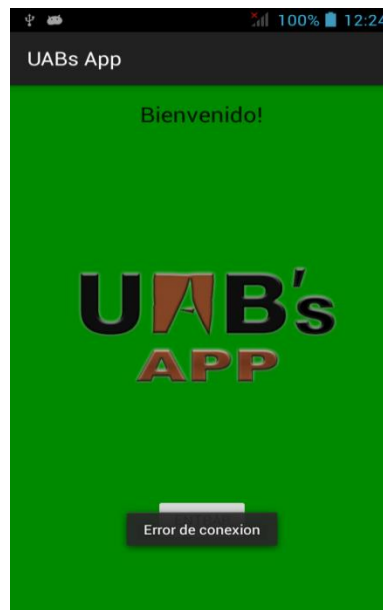


Fig.2: Vista principal con “error de conexión”.

Se debe tener en cuenta que inicialmente la funcionalidad de “*establecerConexion()*” se implementó tomando como referencia un punto de acceso normal ya que todavía no se había creado el punto de acceso que conectaría al servidor y al dispositivo móvil en una misma red.

La segunda parte del aplicativo y la que más trabajo incorpora, es la parte encargada de recibir y enviar mensajes por la red y a su vez poder visualizarlos mediante la interficie gráfica. Esta parte incorpora diversos elementos que hacen muy intuitiva la aplicación para un uso sencillo de la misma. Véase la siguiente imagen: Fig.3.

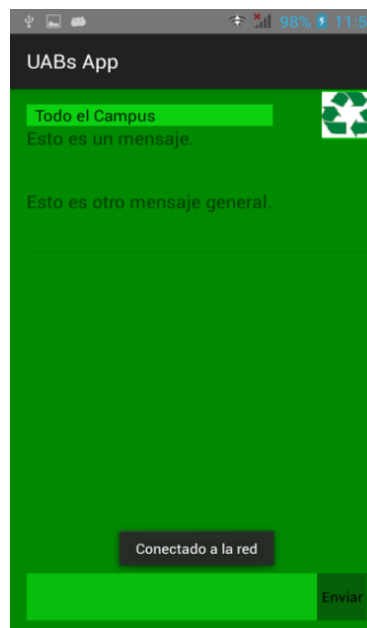


Fig.3: Vista de las salas/grupos del chat del aplicativo.

En primer lugar, se incorpor3 un espacio destinado a la muestra de los mensajes en forma de lista, como es com3nmente conocido en cualquier chat, de forma que los datos recibidos son parseados/filtrados y se muestra la informaci3n necesaria para los usuarios, que en este caso se trata del cuerpo del mensaje. Tambi3n se incorporaron una barra para introducir texto, que servir3 para introducir el mensaje que se quiera comunicar, y un bot3n “Enviar”, el cual iniciar3 el env3o del mensaje hacia el servidor gracias a la funci3n “*enviarDatos()*” que establece una comunicaci3n Http(*Hipertext Transfer Protocol*) con el servidor envi3ndole los datos relacionados con el nuevo mensaje, que son: fecha de creaci3n, identificador del mensaje(ID), cuerpo del mensaje y la facultad a la que se hace referencia.

Otro elemento que se a3adi3 a la segunda vista del aplicativo fue una barra desplegable que contiene el nombre de las distintas facultades que se encuentran en el Campus, de forma que al pulsar en alguna de ellas se mostrar3n los mensajes con relaci3n a dicha facultad, como se observa en la siguiente imagen: Fig.4.

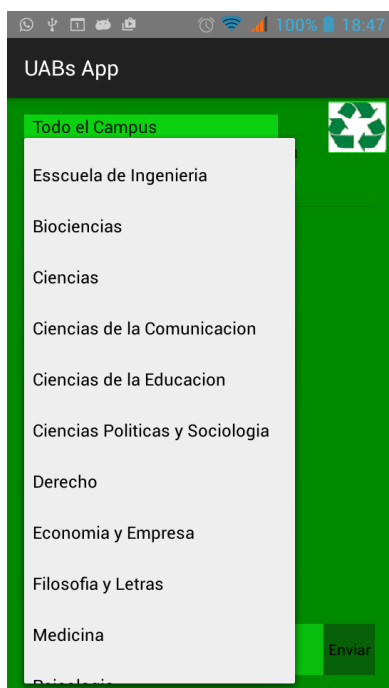


Fig.4: Barra desplegable de “facultades”.

Finalmente, y despu3s de barajar diversas posibilidades, se introdujo un bot3n el cual iniciar3 una funci3n llamada “*obtenerDatos()*” la cual se encarga de establecer una comunicaci3n Http con el servidor para que este le env3e todos los mensajes que se encuentran en su base de datos. Esta era una de las posibilidades para obtener los datos que contiene el servidor, tambi3n se podr3a haber realizado con un contador de tiempo de manera que cada cierto intervalo de tiempo se llamase a la funci3n “*obtenerDatos()*” para ir actualizando la lista de mensajes, pero se opt3 por incorporar un bot3n para hacerlo m3s sencillo. Cabe destacar que la implementaci3n de las funciones

“*obtenerDatos()*” y “*enviarDatos()*” se ha podido realizar correctamente gracias a la incorporaci3n de una librer3a, que ha sido desarrollada sobre la librer3a de Apache “*HttpClient*” para *Android*, y que permite realizar llamadas Http de forma as3ncrona y que se utiliza en aplicaciones con un gran n3mero de descargas como pueden ser *Instagram*, *Pinterest*, etc [1].

4.2.2 Desarrollo de la infraestructura de red

Como ya se ha comentado en otros apartados de este documento, el proyecto se basa en un escenario de red en concreto que debe ser creado para que el aplicativo pueda funcionar sobre dicho escenario.

El primer objetivo que presenta la elaboraci3n del escenario es el de crear un punto de acceso para que los dispositivos m3viles puedan propagar informaci3n por la red y tambi3n recibir informaci3n de la misma. La realizaci3n de este objetivo se podr3a llevar a cabo de diversas formas, pero en este proyecto en concreto se ha utilizado la aplicaci3n “*Virtual Router - WiFi Hot Spot*” la cual nos permite crear un punto de acceso de forma sencilla en una m3quina con sistema operativo(SO) *Windows 7*, *Windows 8* o *Windows Server 2008 R2* [2]. Se decidi3 utilizar este software, ya que la m3quina sobre la que se realiza el proyecto contiene el SO *Windows 8* y adem3s se trata de un software de c3digo libre hospedado en *CodePlex* con el soporte de *Microsoft*. Este software implementa WPA2 como m3todo de encriptaci3n *wireless* y nos permite configurar el SSID y la contrase3a del punto de acceso(v3ase la imagen A1 del ap3ndice).

El segundo objetivo que hab3a que solventar para la elaboraci3n del escenario era la creaci3n de un servidor el cual se pudiese comunicar con los dispositivos m3viles y a la vez pudiese almacenar los datos que se precisan. Para realizar este objetivo se opt3 por utilizar la infraestructura de internet LAMP. LAMP es un acr3nimo que responde al conjunto de herramientas usadas para definir la infraestructura de un servidor web que son: *Linux* como SO, *Apache* como servidor web, *MySQL* como gestor de bases de datos y *PHP*(*Hipertext Processor*) como lenguaje de programaci3n. Se decidi3 usar LAMP en el servidor ya que se trata de herramientas de c3digo libre que han sido muy utilizadas y de la cual hay mucha informaci3n al respecto. Como solo se dispon3a de una sola m3quina para la realizaci3n del proyecto, lo que se hizo fue crear una m3quina virtual, dentro de la m3quina anfitr3n, con el SO *Ubuntu* y hacer que dicha m3quina estuviese conectada virtualmente a la red creada, explicada al principio de este apartado. Posteriormente se procedi3 a la instalaci3n de las herramientas restantes para el correcto funcionamiento del servidor, en este orden: servidor *Apache*, gestor de bases de datos *MySQL* y finalmente el m3dulo *PHP* para procesar c3digo en el servidor y as3 poder extraer de la base de datos la informaci3n solicitada, en nuestro caso, por los dispositivos m3viles.

Una vez instaladas todas las herramientas necesarias, el

siguiente paso fue crear una base de datos acorde con las necesidades de nuestro proyecto, como podemos ver en la Fig.5 la base de datos cuenta con una única tabla donde se guarda la información. La tabla “mensajes” contiene los siguientes atributos: pos (identificador secuencial con respecto a la creación de una nueva entrada en la tabla), id (identificador único para cada mensaje), mensaje (cuerpo del mensaje), fecha (fecha de creación del mensaje) y finalmente facultad (facultad a la cual hace referencia el mensaje).

```
mysql> use uabsapp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> describe mensajes;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| pos   | int(11) | NO | PRI | NULL | auto_increment |
| id    | int(11) | NO |     | NULL |                 |
| mensaje | varchar(100) | YES |     | NULL |                 |
| fecha | date | YES |     | NULL |                 |
| facultad | varchar(50) | YES |     | NULL |                 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)

mysql>
```

Fig.5: Contenido de la tabla “mensajes”

Después de tener la base de datos lista para su uso se procedió a la elaboración de los comportamientos de respuesta que debía realizar el servidor ante las posibles peticiones. En nuestro caso, solo hay 2 posibles demandas que pueden tener los dispositivos móviles con respecto al servidor, que son: petición de todos los datos de la base de datos, o lo que es lo mismo, la petición de todos los mensajes; y la petición de registrar un nuevo mensaje en la base de datos(BD). Para gestionar ambas funcionalidades, los dispositivos móviles a la hora de establecer la comunicación Http con el servidor envían información adicional para que el servidor interprete que se quiere hacer. Una de las variables que se le envía al servidor es la variable “opción”, la cual dependiendo de su valor (0 o 1) dirá si quiere obtener datos de la BD(opción=0) o si quiere introducir datos en la BD (opción=1). Al hacer la petición de los datos al servidor, este los encapsula en un formato JSON para que su trato de sea fácil de gestionar cuando llegan al dispositivo móvil. En el caso de que se quiera introducir datos en la BD, además de la variable “opción”, se deberá enviar el resto de datos necesarios para añadir una nueva entrada en la BD: id, mensaje, fecha y facultad (el atributo “pos” es autogenerado por la propia BD).

A continuación, en la Fig.6, se pueden ver los distintos elementos que forman la red.

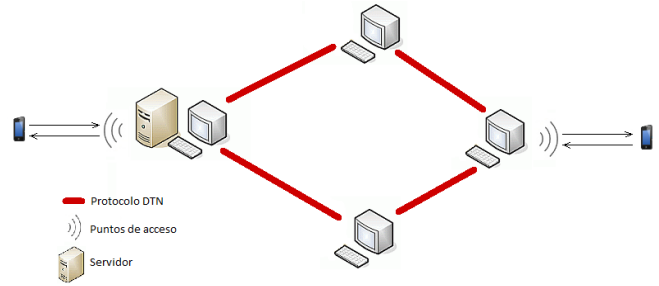


Fig.6: Componentes de la red.

4.3 Fase de test

Cabe destacar que a medida que se iban introduciendo o cumpliendo objetivos, estos eran testeados de manera que se cumplieran su función de la mejor manera posible.

Las distintas funcionalidades nombradas en el apartado anterior no han sido realizadas en el orden en el que aparece cada apartado.

En primer lugar se realizó la parte del aplicativo móvil el cual establecía conexión con un punto de acceso diferente al punto de acceso que se creó finalmente, además se desarrolló la parte que lista los mensajes y que permite introducir nuevos, todo ello de forma local en el propio dispositivo, simplemente para ver la composición de la interficie gráfica ya que por el momento no se tenía ningún tipo de infraestructura de red.

Una vez implementado el punto de acceso en la máquina anfitrión, se comprobó que la aplicación se conectaba correctamente a la red que se le especificaba, mediante el SSID y la contraseña. En una primera instancia todo parecía correcto y la funcionalidad de “establecerConeion()” respondía correctamente. El problema surgió al ver que si des de un inicio el punto de acceso no se encuentra en la lista de redes del dispositivo, aún no ha accedido nunca al punto de acceso creado, la conexión no llegaba a realizarse y el dispositivo se conectaba a otra conexión válida si le era posible. La única solución que se encontró a este problema después de muchas pruebas fue que para que la conexión se estableciera correctamente, previamente se debía haber establecido conexión con el punto de acceso de forma manual, es decir, des de los propios ajustes del dispositivo, entonces sí que se establecía la conexión, estuviera el terminal conectado a una red o no. En el caso de que el terminal esté conectado a otra red distinta, al llamar a la función “establecerConexion()” esta desconecta el dispositivo de la conexión actual y lo reconecta a la red especificada.

Al tener el servidor Apache listo en la máquina virtual y está conectada al punto de acceso creado desde la máquina anfitrión, se realizaron una serie de pruebas para comprobar que la aplicación establecía una comunicación con el servidor. Inicialmente se probó que la aplicación obtuviera la página por defecto del servidor, que en este caso era index.php y solo contenía una línea de texto plano. Dentro de la aplicación no se llegaba a establecer la comunicación, y por consiguiente, no recibía nada del servidor. Esnifando el tráfico de la red mediante la aplica-

ción *Wireshark*, no había ningún paquete que tuviese como dirección de origen la dirección del terminal móvil. El acceso manual desde un navegador al servidor desde el dispositivo móvil sí que establecía la comunicación con lo cual se descartó que el problema estuviese en la parte del servidor o en la infraestructura de red creada. Buscando soluciones por Internet se encontró una librería para *Android* que permitía realizar conexiones Http de forma asíncrona, como ya se ha comentado anteriormente. Al incorporar la nueva librería y utilizar sus funciones para establecer la comunicación con el servidor se obtuvo el resultado esperado y con lo cual se optó por mantener dicha librería.

Con la base de datos ya correctamente instalada en el servidor, se empezaron a realizar las diversas pruebas de inserción y de consulta de datos. En primer lugar se vio que la obtención de los datos se producía de forma desordenada ya que en una primera instancia no existía el atributo "pos" dentro de la tabla "mensajes" de la BD, y el orden de los elementos venía marcado por el atributo "id", que se genera de forma aleatoria por cada mensaje creado en los dispositivos móviles. Al introducir el atributo "pos" que se autogenera al crear una nueva entrada en la BD y de forma incremental, el problema del orden de los mensajes fue solventado. Otro problema que surgió al realizar pruebas de test sobre la BD, fue que la variable "fecha" que se le enviaba al servidor al querer introducir un nuevo mensaje no se filtraba de forma correcta con el formato que *MySQL* contempla y por lo tanto establecía cada entrada con el campo "fecha" con el valor: 0000-00-00. Al realizar una adaptación de la fecha generada desde el dispositivo y enviada con el formato aceptado por *MySQL* el problema quedó solventado.

Por lo que respecta al test de la interfaz gráfica del aplicativo, no generó ningún error de importancia y todos los problemas que surgían, en su gran mayoría se trataba de problemas de codificación, eran testeados y resueltos en el mismo momento de la elaboración de la interfaz.

5 RESULTADOS

Los resultados obtenidos al final del proyecto han sido buenos a grandes rasgos. Pese a que no se ha podido realizar completamente los objetivos marcados inicialmente en el proyecto, se ha podido crear una aplicación que cumple la funcionalidad deseada. La prioridad de los objetivos se marcó inicialmente, en función a la dificultad que conlleva la realización de cada objetivo, y no ha habido ninguna gran modificación al respecto, tan solo una pequeña modificación y es que se marcó como último objetivo desarrollar la interfaz gráfica del aplicativo pero una vez se realizó el primer objetivo se decidió con el tutor del proyecto que la interfaz gráfica que se había desarrollado ya era lo suficientemente válida para no desarrollarla más.

Por otro lado, el hecho de que no se haya podido crear y añadir a la red, unas cuantas máquinas más y que entre ellas (servidor y máquinas de la red) se comunicasen utilizando un protocolo tolerante a retardos ha supuesto una frustración, ya que se trataba de uno de los puntos

que me motivaron a realizar este proyecto entre muchos otros.

6 CONCLUSIONES Y LÍNEAS DE CONTINUACIÓN

A continuación se expondrán una serie de conclusiones extraídas a raíz de la elaboración de este proyecto y también se marcarán unas líneas de continuación para la realización de un trabajo futuro sobre el proyecto.

6.1 Conclusiones

Después de llevar a cabo este proyecto he podido extraer unas cuantas conclusiones al respecto.

La primera conclusión a la que he llegado después de estos 4 meses de trabajo es que en muchas ocasiones la planificación de trabajo se puede ver mermada por imprevistos que surgen a lo largo del desarrollo del trabajo. Obviamente esta conclusión hace referencia a que no se ha podido solventar el último objetivo marcado.

Dado de una parte del proyecto consistía en realizar una aplicación para un dispositivo móvil *Android* he podido aprender la gran cantidad de información que existe al respecto de esta tecnología y la gran comunidad que existe detrás de ella. Las herramientas necesarias para cumplir esta parte, como por ejemplo el entorno de programación, están a disposición de todo usuario que las precise, además con una serie de manuales muy didácticos y completos. El autoaprendizaje en este aspecto ha sido sencillo y muy gratificante, en la gran mayoría de casos.

Otra conclusión a la que he llegado, y la cual está muy relacionada con la conclusión anterior, es que la utilización de herramientas de código libre es siempre un camino a seguir, también por los motivos nombrados anteriormente: mucha información al respecto, una gran comunidad de usuarios que comparte su conocimiento, etc. Además he podido comprobar como el software de código libre permite que los usuarios lo mejoren y que se aprovechen con total garantía de los servicios prestados, haciendo que otros también se aprovechen de las mejoras desarrolladas y formando así una cadena "solidaria", en el desarrollo de la tecnología.

Gracias al desarrollo de este trabajo he podido averiguar que pese a que los dispositivos *Android* incorporan la tecnología DTN en sus terminales, las compañías que los distribuyen la tienen "inaccesible" para los usuarios estándar, y su utilización para usuarios expertos es muy complicada. Esto me lleva a pensar el potencial que puede tener esta tecnología, ya que un buen desarrollo de la misma podría significar, por ejemplo, que muchos usuarios no se vean obligados a contratar servicios de datos con las compañías telefónicas. En referencia a lo comentado anteriormente, un usuario en una ciudad, con el número de dispositivos que eso implica, podría utilizar muchas aplicaciones que no necesiten una comunicación inmediata pero que al fin y al cabo podrían cumplir su función, siempre teniendo en cuenta las dificultades que esta tecnología presenta, como pueden ser: los algoritmos de encaminamiento, que se pueden volver muy complejos, o la seguridad de las comunicaciones.

6.2 Líneas de continuación

A pesar de haber cumplido la gran parte de objetivos marcados desde el inicio del proyecto, han quedado una serie de objetivos sin completar y que considero de vital relevancia.

La agregación de más elementos a la infraestructura de red creada, y que dichos elementos se comuniquen con un protocolo tolerante a retardos, es un objetivo que se puede llevar a cabo después de finalizar este proyecto.

El desarrollo de este trabajo futuro no conlleva excesivo esfuerzo ya que gracias al grupo de investigadores y de docentes SeNDA (*Security of Networks and Distributed Applications*) del departamento de Ingeniería de la Información y de la Comunicación de la Universidad Autónoma de Barcelona, tenemos a disposición una herramienta que se puede incorporar en dispositivos con un SO Linux, y lo dotan de la tecnología necesaria para trabajar en una red tolerante a retardos. De esta forma podríamos cumplir el objetivo que no se ha alcanzado durante la elaboración del proyecto.

AGRADECIMIENTOS

Agradecer a Guillermo Navarro su trabajo como tutor de este proyecto y al departamento de Ingeniería de la Información y la Comunicación por su disposición a ayudar en todo aquello que les es posible.

También quería dar las gracias a mis padres, por su apoyo y porque sin su trabajo nunca habría podido cursar una carrera universitaria en los tiempos que corren.

Finalmente, me gustaría dar las gracias a todos los compañeros, ahora amigos, que he conocido a lo largo del desarrollo de la carrera universitaria y que han sido decisivos a la hora de afrontar muchos trabajos.

BIBLIOGRAFÍA

- [1] James Smith, "Android Asynchronous Http Client" [consultado el 22 de Mayo del 2015] <http://loopj.com/android-async-http/>
- [2] CodePlex, VirtualRouter - WiFi Hot spot [consultado el 11 de Mayo del 2015] <https://virtualrouter.codeplex.com/> . 2013
- [3] SeNDA, ActiveDTN [consultado el 4 de Marzo del 2015] <http://senda.uab.es/Active-DTN>.
- [4] Google Inc, Android Developers [consultado el 18 de Marzo] <http://developer.android.com/index.html>
- [5] Vogella, "Introduction to Android Development with Android Studio" [consultado el 18 de Marzo] <http://www.vogella.com/tutorials/Android/article.html>
- [6] L. Romero Amondaray, "Las redes tolerantes al retardo, una solución a las comunicaciones rurales en Cuba" [consultado el 5 de Marzo] <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/914/2274>
- [7] RFC4838, "Delay-Tolerant Networking Architecture" [consultado el 6 de Marzo] <https://tools.ietf.org/html/rfc4838>
- [8] The PHP Group, "Manual de PHP" [consultado el 21 de Mayo] <http://php.net/manual/es/>
- [9] Oracle, "MySQL Reference Manuals" [consultado el 18 de Mayo] <http://dev.mysql.com/doc/>

APÉNDICE

A1. PANEL DE LA APLICACIÓN VIRTUAL ROUTER

